

- [89] T. Jackson, *On the Design, Implications, and Effects of Implementing Software Diversity for Security*. PhD thesis, University of California, Irvine, 2012.
- [90] R. El-Khalil and A. D. Keromytis, “Hydan: Hiding information in program binaries,” in *Information and Communications Security* (J. Lopez, S. Qing, and E. Okamoto, eds.), (Berlin, Heidelberg), pp. 187–199, Springer Berlin Heidelberg, 2004.
- [91] V. Singhal, A. A. Pillai, C. Saumya, M. Kulkarni, and A. Machiry, “Cornucopia: A framework for feedback guided generation of binaries,” in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, ASE ’22*, (New York, NY, USA), Association for Computing Machinery, 2023.
- [92] B. Cox, D. Evans, A. Filipi, J. Rowanhill, W. Hu, J. Davidson, J. Knight, A. Nguyen-Tuong, and J. Hiser, “N-variant systems: a secretless framework for security through diversity,” in *Proc. of USENIX Security Symposium*, USENIX-SS’06, 2006.
- [93] D. Bruschi, L. Cavallaro, and A. Lanzi, “Diversified process replicas for defeating memory error exploits,” in *Proc. of the Int. Performance, Computing, and Communications Conference*, 2007.
- [94] B. Salamat, A. Gal, T. Jackson, K. Manivannan, G. Wagner, and M. Franz, “Stopping buffer overflow attacks at run-time: Simultaneous multi-variant program execution on a multicore processor,” tech. rep., Technical Report 07-13, School of Information and Computer Sciences, UC Irvine, 2007.
- [95] G. Agosta, A. Barengi, G. Pelosi, and M. Scandale, “The MEET approach: Securing cryptographic embedded software against side channel attacks,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 8, pp. 1320–1333, 2015.
- [96] T. Jackson, B. Salamat, A. Homescu, K. Manivannan, G. Wagner, A. Gal, S. Brunthaler, C. Wimmer, and M. Franz, “Compiler-generated software diversity,” in *Moving Target Defense*, pp. 77–98, Springer, 2011.
- [97] A. Amarilli, S. Müller, D. Naccache, D. Page, P. Rauzy, and M. Tunstall, “Can code polymorphism limit information leakage?,” in *IFIP International Workshop on Information Security Theory and Practices*, pp. 1–21, Springer, 2011.
- [98] A. Voulimeneas, D. Song, P. Larsen, M. Franz, and S. Volckaert, “dmvx: Secure and efficient multi-variant execution in a distributed setting,” in *Proceedings of the 14th European Workshop on Systems Security*, pp. 41–47, 2021.

- [99] I. Bow, N. Bete, F. Saqib, W. Che, C. Patel, R. Robucci, C. Chan, and J. Plusquellic, “Side-channel power resistance for encryption algorithms using implementation diversity,” *Cryptography*, vol. 4, no. 2, 2020.
- [100] R. L. Castro, C. Schmitt, and G. D. Rodosek, “Armed: How automatic malware modifications can evade static detection?,” in *2019 5th International Conference on Information Management (ICIM)*, pp. 20–27, 2019.
- [101] J. Cabrera Arteaga, O. Floros, O. Vera Perez, B. Baudry, and M. Monperrus, “Crow: code diversification for webassembly,” in *MADWeb, NDSS 2021*, 2021.
- [102] R. Sasnauskas, Y. Chen, P. Collingbourne, J. Ketema, G. Lup, J. Taneja, and J. Regehr, “Souper: A Synthesizing Superoptimizer,” *arXiv preprint 1711.04422*, 2017.
- [103] J. Cabrera Arteaga, P. Laperdrix, M. Monperrus, and B. Baudry, “Multi-Variant Execution at the Edge,” *arXiv e-prints*, p. arXiv:2108.08125, Aug. 2021.
- [104] J. Lettner, D. Song, T. Park, P. Larsen, S. Volckaert, and M. Franz, “Partisan: fast and flexible sanitization via run-time partitioning,” in *International Symposium on Research in Attacks, Intrusions, and Defenses*, pp. 403–422, Springer, 2018.
- [105] B. G. Ryder, “Constructing the call graph of a program,” *IEEE Transactions on Software Engineering*, no. 3, pp. 216–226, 1979.
- [106] S. Narayan, C. Disselkoen, D. Moghimi, S. Cauligi, E. Johnson, Z. Gang, A. Vahldiek-Oberwagner, R. Sahita, H. Shacham, D. Tullsen, *et al.*, “Swivel: Hardening webassembly against spectre,” in *USENIX Security Symposium*, 2021.
- [107] E. Johnson, D. Thien, Y. Alhessi, S. Narayan, F. Brown, S. Lerner, T. McMullen, S. Savage, and D. Stefan, “Sfi safety for native-compiled wasm,” *NDSS. Internet Society*, 2021.
- [108] J. Cabrera-Arteaga, N. Fitzgerald, M. Monperrus, and B. Baudry, “WASM-MUTATE: Fast and Effective Binary Diversification for WebAssembly,” *arXiv e-prints*, p. arXiv:2309.07638, Sept. 2023.
- [109] M. Willsey, C. Nandi, Y. R. Wang, O. Flatt, Z. Tatlock, and P. Panchekha, “Egg: Fast and extensible equality saturation,” *Proc. ACM Program. Lang.*, vol. 5, jan 2021.
- [110] “Stop a wasm compiler bug before it becomes a problem | fastly.” <https://www.fastly.com/blog/defense-in-depth-stopping-a-wasm-compiler-bug-before-it-became-a-problem>, 2021.